1

码垛工作站案例分析

1、IO 单元信号定义 DSQC652 IO 单元

2、系统信号的关联

3、程序数据

PERS wobjdata WobjPallet\_L:=[FALSE,TRUE,"",[[-456.216,-2058.49,-233.373],[1,0,0,0]],[[0,0,0],[1,0,0,0]]];

PERS wobjdata WobjPallet\_R:=[FALSE,TRUE,"",[[-421.764,1102.39,-233.373],[1,0,0,0]],[[0,0,0],[1,0,0,0]]];

PERS tooldata tGripper:=[TRUE,[[0,0,527],[1,0,0,0]],[20,[0,0,150],[1,0,0,0],0,0,0]];

PERS loaddata LoadFull:=[20,[0,0,300],[1,0,0,0],0,0,0.1];

PERS wobjdata CurWobj;

PERS jointtarget jposHome:=[[0,0,0,0,0,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarget

pPlaceBase0\_L:=[[296.473529255,212.21064316,3.210904169],[0,0.70711295,-0.707100612,0],[-2,0,-3,0],[9E9,9E9,9E9,9E9,9E9,9E9]];

CONST robtarget

pPlaceBase90\_L:=[[218.407102669,695.953395421,3.210997808],[0,-0.000001669,1,0],[-2,0,-2,0],[9E9,9E9,9E9,9E9,9E9,9E9]];

CONST robtarget



2

pPlaceBase0\_R:=[[296.473529255,212.21064316,3.210904169],[0,0.707221603,-0.70699194,0],[1,0,0,0],[9E9,9E9,9E9,9E9,9E9,9E9]];

CONST robtarget

pPlaceBase90\_R:=[[218.407102669,695.953395421,3.210997808],[0,-0.00038594,0.999999926,0],[1,0,1,0],[9E9,9E9,9E9,9E9,9E9,9E9]];

CONST robtarget

pPick\_L:=[[1627.550991372,-426.974661352,-26.736921885],[0,0.707109873,-0.707103689,0],[-1,0,-2,0],[9E9,9E9,9E9,9E9,9E9,9E9]];

CONST robtarget

pPick\_R:=[[1611.055992534,442.364097921,-26.736584068],[0,0.707220363,-0.706993181,0],[0,0,-1,0],[9E9,9E9,9E9,9E9,9E9,9E9]];

CONST robtarget

pHome:=[[1505.00,-0.00,878.55],[1.28548E-06,0.707107,-0.707107,-1.26441E-06],[0,0,-2,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

PERS robtarget pPlaceBase0;

PERS robtarget pPlaceBase90;

PERS robtarget pPick;

PERS robtarget pPlace;

PERS robtarget pPickSafe;

PERS num nCycleTime:=3.263;

PERS num nCount\_L:=9;

PERS num nCount\_R:=13;

PERS num nPallet:=1;

PERS num nPalletNo:=2;

PERS num nPickH:=300;

PERS num nPlaceH:=400;

PERS num nBoxL:=605;

PERS num nBoxW:=405;

PERS num nBoxH:=300;

VAR clock Timer1;

PERS bool bReady:=TRUE;

PERS bool bPalletFull\_L:=FALSE;

PERS bool bPalletFull\_R:=FALSE;

PERS bool bGetPosition:=TRUE;

VAR triggdata HookAct;

VAR triggdata HookOff;

VAR intnum iPallet\_L;

VAR intnum iPallet\_R;

PERS speeddata vMinEmpty:=[2000,400,6000,1000];

PERS speeddata vMidEmpty:=[3000,400,6000,1000];

PERS speeddata vMaxEmpty:=[5000,500,6000,1000];

PERS speeddata vMinLoad:=[1000,200,6000,1000];

PERS speeddata vMidLoad:=[2500,500,6000,1000];

PERS speeddata vMaxLoad:=[4000,500,6000,1000];

PERS num

Compensation{15,3}:=[[0,0,0],[0,0,0],[0,0,0],[0,0,0],[0,0,0],[0,0,0],[0,0,0],[0,0,0],[0,0,0],[0,0,0],[0,0,0],[0,0,0],[0,0,0],[0,0,0],[0,0,0]];

4.主程序

PROC main()

rInitAll;

WHILE TRUE DO

IF bReady THEN

rPick;

rPlace;

ENDIF

rCycleCheck;



3

ENDWHILE

ENDPROC

5.其他例行程序

PROC rInitAll()

rCheckHomePos;

ConfL\OFF;

ConfJ\OFF;

nCount\_L:=1;

nCount\_R:=1;

nPallet:=1;

nPalletNo:=1;

bPalletFull\_L:=FALSE;

bPalletFull\_R:=FALSE;

bGetPosition:=FALSE;

Reset do00\_ClampAct;

Reset do01\_HookAct;

ClkStop Timer1;

ClkReset Timer1;

TriggEquip HookAct,100,0.1\DOp:=do01\_HookAct,1;

TriggEquip HookOff,100\Start,0.1\DOp:=do01\_HookAct,0;

IDelete iPallet\_L;

CONNECT iPallet\_L WITH tEjectPallet\_L;

ISignalDI di02\_PalletInPos\_L,0,iPallet\_L;

IDelete iPallet\_R;

CONNECT iPallet\_R WITH tEjectPallet\_R;

ISignalDI di03\_PalletInPos\_R,0,iPallet\_R;

ENDPROC

PROC rPick()

ClkReset Timer1;

ClkStart Timer1;

rCalPosition;

MoveJ Offs(pPick,0,0,nPickH),vMaxEmpty,z50,tGripper\WObj:=wobj0;

MoveL pPick,vMinLoad,fine,tGripper\WObj:=wobj0;

Set do00\_ClampAct;

Waittime 0.3;

GripLoad LoadFull;

TriggL Offs(pPick,0,0,nPickH),vMinLoad,HookAct,z50,tGripper\WObj:=wobj0;

MoveL pPickSafe,vMaxLoad,z100,tGripper\WObj:=wobj0;

ENDPROC

PROC rPlace()

MoveJ Offs(pPlace,0,0,nPlaceH),vMaxLoad,z50,tGripper\WObj:=CurWobj;

TriggL pPlace,vMinLoad,HookOff,fine,tGripper\WObj:=CurWobj;

Reset do00\_ClampAct;

Waittime 0.3;

GripLoad Load0;

MoveL Offs(pPlace,0,0,nPlaceH),vMinEmpty,z50,tGripper\WObj:=CurWobj;

rPlaceRD;

MoveJ pPickSafe,vMaxEmpty,z50,tGripper\WObj:=wobj0;



4

ClkStop Timer1;

nCycleTime:=ClkRead(Timer1);

ENDPROC

PROC rCycleCheck()

TPErase;

TPWrite "The Robot is running!";

TPWrite "Last cycle time is : "\Num:=nCycleTime;

TPWrite "The number of the Boxes in the Left pallet is:"\Num:=nCount\_L-1;

TPWrite "The number of the Boxes in the Right pallet is:"\Num:=nCount\_R-1;

IF (bPalletFull\_L=FALSE AND di02\_PalletInPos\_L=1 AND di00\_BoxInPos\_L=1) OR (bPalletFull\_R=FALSE AND di03\_PalletInPos\_R=1

AND di01\_BoxInPos\_R=1) THEN

bReady:=TRUE;

ELSE

bReady:=FALSE;

WaitTime 0.1;

ENDIF

ENDPROC

PROC rCalPosition()

bGetPosition:=FALSE;

WHILE bGetPosition=FALSE DO

TEST nPallet

CASE 1:

IF bPalletFull\_L=FALSE AND di02\_PalletInPos\_L=1 AND di00\_BoxInPos\_L=1 THEN

pPick:=pPick\_L;

pPlaceBase0:=pPlaceBase0\_L;

pPlaceBase90:=pPlaceBase90\_L;

CurWobj:=WobjPallet\_L;

pPlace:=pPattern(nCount\_L);

bGetPosition:=TRUE;

nPalletNo:=1;

ELSE

bGetPosition:=FALSE;

ENDIF

nPallet:=2;

CASE 2:

IF bPalletFull\_R=FALSE AND di03\_PalletInPos\_R=1 AND di01\_BoxInPos\_R=1 THEN

pPick:=pPick\_R;

pPlaceBase0:=pPlaceBase0\_R;

pPlaceBase90:=pPlaceBase90\_R;

CurWobj:=WobjPallet\_R;

pPlace:=pPattern(nCount\_R);

bGetPosition:=TRUE;

nPalletNo:=2;

ELSE

bGetPosition:=FALSE;

ENDIF

nPallet:=1;

DEFAULT:



5

TPERASE;

TPWRITE "The data 'nPallet' is error,please check it!";

Stop;

ENDTEST

ENDWHILE

ENDPROC

FUNC robtarget pPattern(num nCount)

VAR robtarget pTarget;

IF nCount>=1 AND nCount<=5 THEN

pPickSafe:=Offs(pPick,0,0,400);

ELSEIF nCount>=6 AND nCount<=10 THEN

pPickSafe:=Offs(pPick,0,0,600);

ELSEIF nCount>=11 AND nCount<=15 THEN

pPickSafe:=Offs(pPick,0,0,800);

ENDIF

TEST nCount

CASE 1:

pTarget.trans.x:=pPlaceBase0.trans.x;

pTarget.trans.y:=pPlaceBase0.trans.y;

pTarget.trans.z:=pPlaceBase0.trans.z;

pTarget.rot:=pPlaceBase0.rot;

pTarget.robconf:=pPlaceBase0.robconf;

pTarget:=Offs(pTarget,Compensation{nCount,1},Compensation{nCount,2},Compensation{nCount,3});

CASE 2:

pTarget.trans.x:=pPlaceBase0.trans.x+nBoxL;

pTarget.trans.y:=pPlaceBase0.trans.y;

pTarget.trans.z:=pPlaceBase0.trans.z;

pTarget.rot:=pPlaceBase0.rot;

pTarget.robconf:=pPlaceBase0.robconf;

pTarget:=Offs(pTarget,Compensation{nCount,1},Compensation{nCount,2},Compensation{nCount,3});

CASE 3:

pTarget.trans.x:=pPlaceBase90.trans.x;

pTarget.trans.y:=pPlaceBase90.trans.y;

pTarget.trans.z:=pPlaceBase90.trans.z;

pTarget.rot:=pPlaceBase90.rot;

pTarget.robconf:=pPlaceBase90.robconf;

pTarget:=Offs(pTarget,Compensation{nCount,1},Compensation{nCount,2},Compensation{nCount,3});

CASE 4:

pTarget.trans.x:=pPlaceBase90.trans.x+nBoxW;

pTarget.trans.y:=pPlaceBase90.trans.y;

pTarget.trans.z:=pPlaceBase90.trans.z;

pTarget.rot:=pPlaceBase90.rot;

pTarget.robconf:=pPlaceBase90.robconf;

pTarget:=Offs(pTarget,Compensation{nCount,1},Compensation{nCount,2},Compensation{nCount,3});

CASE 5:

pTarget.trans.x:=pPlaceBase90.trans.x+2\*nBoxW;

pTarget.trans.y:=pPlaceBase90.trans.y;

pTarget.trans.z:=pPlaceBase90.trans.z;

pTarget.rot:=pPlaceBase90.rot;



6

pTarget.robconf:=pPlaceBase90.robconf;

pTarget:=Offs(pTarget,Compensation{nCount,1},Compensation{nCount,2},Compensation{nCount,3});

CASE 6:

pTarget.trans.x:=pPlaceBase0.trans.x;

pTarget.trans.y:=pPlaceBase0.trans.y+nBoxL;

pTarget.trans.z:=pPlaceBase0.trans.z+nBoxH;

pTarget.rot:=pPlaceBase0.rot;

pTarget.robconf:=pPlaceBase0.robconf;

pTarget:=Offs(pTarget,Compensation{nCount,1},Compensation{nCount,2},Compensation{nCount,3});

CASE 7:

pTarget.trans.x:=pPlaceBase0.trans.x+nBoxL;

pTarget.trans.y:=pPlaceBase0.trans.y+nBoxL;

pTarget.trans.z:=pPlaceBase0.trans.z+nBoxH;

pTarget.rot:=pPlaceBase0.rot;

pTarget.robconf:=pPlaceBase0.robconf;

pTarget:=Offs(pTarget,Compensation{nCount,1},Compensation{nCount,2},Compensation{nCount,3});

CASE 8:

pTarget.trans.x:=pPlaceBase90.trans.x;

pTarget.trans.y:=pPlaceBase90.trans.y-nBoxW;

pTarget.trans.z:=pPlaceBase90.trans.z+nBoxH;

pTarget.rot:=pPlaceBase90.rot;

pTarget.robconf:=pPlaceBase90.robconf;

pTarget:=Offs(pTarget,Compensation{nCount,1},Compensation{nCount,2},Compensation{nCount,3});

CASE 9:

pTarget.trans.x:=pPlaceBase90.trans.x+nBoxW;

pTarget.trans.y:=pPlaceBase90.trans.y-nBoxW;

pTarget.trans.z:=pPlaceBase90.trans.z+nBoxH;

pTarget.rot:=pPlaceBase90.rot;

pTarget.robconf:=pPlaceBase90.robconf;

pTarget:=Offs(pTarget,Compensation{nCount,1},Compensation{nCount,2},Compensation{nCount,3});

CASE 10:

pTarget.trans.x:=pPlaceBase90.trans.x+2\*nBoxW;

pTarget.trans.y:=pPlaceBase90.trans.y-nBoxW;

pTarget.trans.z:=pPlaceBase90.trans.z+nBoxH;

pTarget.rot:=pPlaceBase90.rot;

pTarget.robconf:=pPlaceBase90.robconf;

pTarget:=Offs(pTarget,Compensation{nCount,1},Compensation{nCount,2},Compensation{nCount,3});

CASE 11:

pTarget.trans.x:=pPlaceBase0.trans.x;

pTarget.trans.y:=pPlaceBase0.trans.y;

pTarget.trans.z:=pPlaceBase0.trans.z+2\*nBoxH;

pTarget.rot:=pPlaceBase0.rot;

pTarget.robconf:=pPlaceBase0.robconf;

pTarget:=Offs(pTarget,Compensation{nCount,1},Compensation{nCount,2},Compensation{nCount,3});

CASE 12:

pTarget.trans.x:=pPlaceBase0.trans.x+nBoxL;

pTarget.trans.y:=pPlaceBase0.trans.y;

pTarget.trans.z:=pPlaceBase0.trans.z+2\*nBoxH;

pTarget.rot:=pPlaceBase0.rot;

pTarget.robconf:=pPlaceBase0.robconf;



7

pTarget:=Offs(pTarget,Compensation{nCount,1},Compensation{nCount,2},Compensation{nCount,3});

CASE 13:

pTarget.trans.x:=pPlaceBase90.trans.x;

pTarget.trans.y:=pPlaceBase90.trans.y;

pTarget.trans.z:=pPlaceBase90.trans.z+2\*nBoxH;

pTarget.rot:=pPlaceBase90.rot;

pTarget.robconf:=pPlaceBase90.robconf;

pTarget:=Offs(pTarget,Compensation{nCount,1},Compensation{nCount,2},Compensation{nCount,3});

CASE 14:

pTarget.trans.x:=pPlaceBase90.trans.x+nBoxW;

pTarget.trans.y:=pPlaceBase90.trans.y;

pTarget.trans.z:=pPlaceBase90.trans.z+2\*nBoxH;

pTarget.rot:=pPlaceBase90.rot;

pTarget.robconf:=pPlaceBase90.robconf;

pTarget:=Offs(pTarget,Compensation{nCount,1},Compensation{nCount,2},Compensation{nCount,3});

CASE 15:

pTarget.trans.x:=pPlaceBase90.trans.x+2\*nBoxW;

pTarget.trans.y:=pPlaceBase90.trans.y;

pTarget.trans.z:=pPlaceBase90.trans.z+2\*nBoxH;

pTarget.rot:=pPlaceBase90.rot;

pTarget.robconf:=pPlaceBase90.robconf;

pTarget:=Offs(pTarget,Compensation{nCount,1},Compensation{nCount,2},Compensation{nCount,3});

DEFAULT:

TPErase;

TPWrite "The data 'nCount' is error,please check it !";

stop;

ENDTEST

Return pTarget;

ENDFUNC

PROC rPlaceRD()

TEST nPalletNo

CASE 1:

Incr nCount\_L;

IF nCount\_L>15 THEN

Set do02\_PalletFull\_L;

bPalletFull\_L:=TRUE;

nCount\_L:=1;

ENDIF

CASE 2:

Incr nCount\_R;

IF nCount\_R>15 THEN

Set do03\_PalletFull\_R;

bPalletFull\_R:=TRUE;

nCount\_R:=1;

ENDIF

DEFAULT:

TPERASE;

TPWRITE "The data 'nPalletNo' is error,please check it!";

Stop;



8

ENDTEST

ENDPROC

PROC rCheckHomePos()

VAR robtarget pActualPos;

IF NOT CurrentPos(pHome,tGripper) THEN

pActualpos:=CRobT(\Tool:=tGripper\WObj:=wobj0);

pActualpos.trans.z:=pHome.trans.z;

MoveL pActualpos,v500,z10,tGripper;

MoveJ pHome,v1000,fine,tGripper;

ENDIF

ENDPROC

FUNC bool CurrentPos(robtarget ComparePos,INOUT tooldata TCP)

VAR num Counter:=0;

VAR robtarget ActualPos;

ActualPos:=CRobT(\Tool:=TCP\WObj:=wobj0);

IF ActualPos.trans.x>ComparePos.trans.x-25 AND ActualPos.trans.x<ComparePos.trans.x+25 Counter:=Counter+1;

IF ActualPos.trans.y>ComparePos.trans.y-25 AND ActualPos.trans.y<ComparePos.trans.y+25 Counter:=Counter+1;

IF ActualPos.trans.z>ComparePos.trans.z-25 AND ActualPos.trans.z<ComparePos.trans.z+25 Counter:=Counter+1;

IF ActualPos.rot.q1>ComparePos.rot.q1-0.1 AND ActualPos.rot.q1<ComparePos.rot.q1+0.1 Counter:=Counter+1;

IF ActualPos.rot.q2>ComparePos.rot.q2-0.1 AND ActualPos.rot.q2<ComparePos.rot.q2+0.1 Counter:=Counter+1;

IF ActualPos.rot.q3>ComparePos.rot.q3-0.1 AND ActualPos.rot.q3<ComparePos.rot.q3+0.1 Counter:=Counter+1;

IF ActualPos.rot.q4>ComparePos.rot.q4-0.1 AND ActualPos.rot.q4<ComparePos.rot.q4+0.1 Counter:=Counter+1;

RETURN Counter=7;

ENDFUNC

TRAP tEjectPallet\_L

Reset do02\_PalletFull\_L;

bPalletFull\_L:=FALSE;

ENDTRAP

TRAP tEjectPallet\_R

Reset do03\_PalletFull\_R;

bPalletFull\_R:=FALSE;

ENDTRAP

PROC rMoveAbsj()

MoveAbsJ jposHome\NoEOffs, v100, fine, tGripper\WObj:=wobj0;

ENDPROC

PROC rModPos()

MoveL pHome,v100,fine,tGripper\WObj:=Wobj0;

MoveL pPick\_L,v100,fine,tGripper\WObj:=Wobj0;

MoveL pPick\_R,v100,fine,tGripper\WObj:=Wobj0;

MoveL pPlaceBase0\_L,v100,fine,tGripper\WObj:=WobjPallet\_L;

MoveL pPlaceBase90\_L,v100,fine,tGripper\WObj:=WobjPallet\_L;

MoveL pPlaceBase0\_R,v100,fine,tGripper\WObj:=WobjPallet\_R;

MoveL pPlaceBase90\_R,v100,fine,tGripper\WObj:=WobjPallet\_R;

ENDPROC

